A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors

Priyanga Dilini Talagala[1,2], Rob J. Hyndman[1,2], Catherine Leigh[1,3], Kerrie Mengersen[1,4], Kate Smith-Miles[1,5]

[1]ARC Centre of Excellence for Mathematics and Statistical Frontiers (ACEMS), Australia
[2]Department of Econometrics and Business Statistics, Monash University, Australia
[3]Institute for Future Environments, Science and Engineering Faculty, Queensland University of Technology, Australia
[4]School of Mathematical Sciences, Science and Engineering Faculty, Queensland University of Technology, Australia
[5]School of Mathematics and Statistics, University of Melbourne, Australia

Key Points:

- Our feature-based procedure starts by applying different statistical transformations to water-quality data to highlight outliers in high dimensional space
- Density and distance-based unsupervised outlier scoring techniques were applied to detect outliers due to technical issues with the sensors
- An approach based on extreme value theory was then used to calculate outlier thresholds

Corresponding author: Priyanga Dilini Talagala, `dilini.talagala@monash.edu`

Abstract
Outliers due to technical errors in water quality data from *in situ* sensors can reduce data quality and have a direct impact on inference drawn from subsequent data analysis. However, outlier detection through manual monitoring is infeasible given the volume and velocity of data the sensors produce. Here, we introduce an automated procedure, named oddwater that provides early detection of outliers in water-quality data from *in situ* sensors caused by technical issues. Our oddwater procedure is used to first identify the data features that differentiate outlying instances from typical behaviours. Then statistical transformations are applied to make the outlying instances stand out in a transformed data space. Unsupervised outlier scoring techniques are applied to the transformed data space and an approach based on extreme value theory is used to calculate a threshold for each potential outlier. Using two datasets obtained from *in situ* sensors in rivers flowing into the Great Barrier Reef lagoon, Australia, we show that oddwater successfully identifies outliers involving abrupt changes in turbidity, conductivity and river level, including sudden spikes, sudden isolated drops and level shifts, while maintaining very low false detection rates. We have implemented this oddwater procedure in the open source R package `oddwater`.

1 Introduction

Water-quality monitoring traditionally relies on water samples collected manually. The samples are then analyzed within laboratories to determine the water-quality variables of interest. This type of rigorous laboratory analysis of field-collected samples is crucial in making natural resources management decisions that affect human welfare and environmental conditions. However, with the rapid advances in hardware technology, the use of *in situ* water-quality sensors positioned at different geographic sites is becoming an increasingly common practice used to acquire real-time measurements of environmental and water-quality variables. Though only a subset of the required water-quality variables can be measured by these sensors, they have several advantages. Their ability to collect large quantities of data and to archive historic records allows for deeper analysis of water-quality variables to improve understanding about field conditions and water-quality processes (Glasgow et al., 2004). Near-real-time monitoring also allows operators to identify and respond to potential issues quickly and thus manage the operations efficiently. Further, the use of *in situ* sensors can greatly reduce the labor involved in field sampling and laboratory analysis.

Water-quality sensors are exposed to changing environments and extreme weather conditions, and thus are prone to errors, including failure. Automated detection of outliers in water-quality data from *in situ* sensors has therefore captured the attention of many researchers both in the ecology and data science communities (Hill et al., 2009; Archer et al., 2003; Raciti et al., 2012; McKenna et al., 2007; Koch & McKenna, 2010). This problem of outlier detection in water-quality data from *in situ* sensors can be divided into two sub-topics according to their focus: (1) identifying errors in the data due to issues unrelated to water events per se, such as technical aberrations, that make the data unreliable and untrustworthy; and (2) identifying real events (e.g. rare but sudden spikes in turbidity associated with rare but sudden high-flow events). Both problems are equally important when making natural resource management decisions that affect human welfare and environmental conditions. Problem 1 can also be considered as a data preprocessing phase before addressing Problem 2.

In this work we focus on Problem 1, i.e. detecting unusual measurements caused by technical errors that make data unreliable and untrustworthy, and affect performance of any subsequent data analysis under Problem 2. According to Yu (2012), the degree of confidence in the sensor data is one of the main requirements for a properly defined

environmental analysis procedure. For instance, researchers and policy makers are unable to use water-quality data containing technical outliers with confidence for decision making and reporting purposes, because erroneous conclusions regarding the quality of the water being monitored could ensue, leading, for example, to inappropriate or unnecessary water treatment, land management or warning alerts to the public (Kotamäki et al., 2009; Rangeti et al., 2015). Missing values and corrupted data can also have an adverse impact on water-quality model building and calibration processes (Archer et al., 2003). Early detection of these technical outliers will limit the use of corrupted data for subsequent analysis. For instance, it will limit the use of corrupted data in real-time forecasting and online applications such as on-line drinking water-quality monitoring and early warning systems (Storey et al., 2011), predicting algal bloom outbreaks leading to fish kill events and potential human health impacts, forecasting water level and currents etc. (Glasgow et al., 2004; Archer et al., 2003; Hill & Minsker, 2006). However, because data arrive near continuously at high speed in large quantities, manual monitoring is highly unlikely to be able to capture all the errors. These issues have therefore increased the importance of developing automated methods for early detection of outliers in water-quality data from *in situ* sensors (Hill et al., 2009).

Different statistical approaches are available to detect outliers in water-quality data from *in situ* sensors. For example, Hill and Minsker (2006) addressed the problem of outlier detection in environmental sensors using regression-based time series models. In this work they addressed the scenario as a univariate problem. Their prediction models are based on four data-driven methods: naive, clustering, perceptron, and Artificial Neural Networks (ANN). Measurements that fell outside the bounds of an established prediction interval were declared as outliers. They also considered two strategies: anomaly detection (AD) and anomaly detection and mitigation (ADAM) for the detection process. ADAM replaces detected outliers with the predicted value prior to the next predictions whereas AD simply uses the previous measurements without making any alteration to the detected outliers. These types of data-driven methods develop models using sets of training examples containing a feature set and a target output. Later, Hill et al. (2009) addressed the problem by developing three automated anomaly detection methods using dynamic Bayesian networks (DBN) and showed that DBN-based detectors using either robust Kalman filtering or Rao-Blackwellized particle filtering, outperformed that of Kalman filtering.

Another common approach for detecting outliers in environmental sensor data is based on residuals (the differences between predicted and actual values). Due to the ability of ANNs to model a wide range of complex non-linear phenomena, Moatar, Fessant, and Poirel (1999) used ANN techniques to detect anomalies such as abnormal values, discontinuities, and drifts in pH readings. After developing the pH model, the Student t-test and the cumulative Page–Hinkley test were applied to detect changes in the mean of the residuals to detect measurement error occurring over short periods of time. The work was later expanded to a multivariate scenario with some additional water-quality variables including dissolved oxygen, electrical conductivity, pH and temperature (Moatar et al., 2001). Their proposed algorithm used both deterministic and stochastic approaches for the model building process. Observed data were then compared with the model forecasts using a set of classical statistical tests to detect outliers, demonstrating the effectiveness and advantages of the multimodel approach. Later, Archer et al. (2003) proposed a method to detect failures in the water-quality sensors due to biofouling based on a sequential likelihood ratio test. Their method also had the ability to provide estimates of biofouling onset time, which was useful for the subsequent step of outlier correction.

A common feature of all of the above methods is that they are usually employed in a supervised or semi-supervised context and thus require training data pre-labeled with known outliers or data that are free from the anomalous features of interest. In many cases, however, not all the possible outliers are known in advance and can arise spon-

taneously as new outlying behaviors during the test phase. In such situations, supervised methods may fail to detect those outliers. Semi-supervised methods are also unsuitable for certain applications due to the unavailability of training data containing only typical instances that are free from outliers (Goldstein & Uchida, 2016). The datasets that we consider in this paper suffer from both of these limitations highlighting the need for a more general approach.

This paper develops a method for detecting technical outliers in water-quality data derived from *in situ* sensors. Prior work by Leigh et al. (2019) emphasises the importance of different anomaly types and end-user needs and provides the starting point for constructing a framework for automated anomaly detection in high frequency water-quality data from *in situ* sensors. Their work briefly introduced unsupervised feature based methods for detecting technical-outliers in such data. The present paper differs substantially from Leigh et al. (2019) as (1) the unsupervised feature based procedure we present for detecting technical-outliers in high frequency water-quality data measured by *in situ* sensors is its sole focus (2) the unsupervised feature based procedure is fully elaborated in both details and depth and (3) the experimental results are enhanced through emphasis on the multivariate capabilities of the unsupervised feature based procedure. Furthermore, we focus on outliers involving abrupt changes in value, including sudden spikes, sudden isolated drops and level shifts (high priority outliers as described in Leigh et al. (2019)) rather than the broader suite considered by Leigh et al. (2019).

First, we present in detail our unsupervised feature based procedure that provides early detection of technical outliers in water-quality data from *in situ* sensors. Rule-based methods are also incorporated into the procedure to flag occurrences of impossible, out-of-range, and missing values. Second, we provide a comparative analysis of the efficacy and reliability of both density- and nearest neighbor distance-based outlier scoring techniques. Third, we introduce an R (R Core Team, 2018) package, `oddwater` (Talagala & Hyndman, 2018) that implements the feature-based procedure and related functions. Further, to facilitate reproducibility and reuse of the results presented in this paper we have made all of the code and associated datasets available on zenodo[1].

Our feature-based procedure has many advantages: (1) it can take the correlation structure of the water-quality variables into account when detecting outliers; (2) it can be applied to both univariate and multivariate problems; (3) the outlier scoring techniques that we consider are unsupervised, data-driven approaches and therefore do not require training datasets for the model building process, and can be extended easily to other time series from other sites; (4) the outlier thresholds have a probabilistic interpretation as they are based on extreme value theory; (5) the approach has the ability to deal with irregular (unevenly spaced) time series; and (6) it can easily be extended to streaming data. In contrast to a batch scenario, which assumes that the entire dataset is available prior to the analysis with the focus on detecting complete events, the streaming data scenario gives many additional challenges due to high velocity, unbounded, nonstationary data with incomplete events (Hill et al., 2009; Talagala et al., 2018). In this paper, although our oddwater procedure is introduced as a batch method it can easily be extended to streaming data such that it can provide near-real-time support using a sliding window technique.

2 Materials and Methods

Our unsupervised feature-based procedure for detecting outliers in water-quality data from *in situ* sensors has six main steps (Figure 1), and the structure of this section is organised accordingly. For easy reference, we named our unsupervised feature-based

[1] DOI: 10.5281/zenodo.2890469

procedure as oddwater procedure, which stands for **O**utlier **D**etection in **D**ata from **WA-TER**-quality sensors

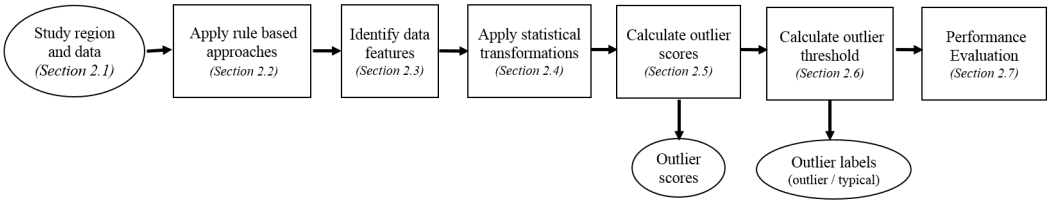

Figure 1. Unsupervised feature-based procedure, named oddwater procedure for outlier detection in water quality data from *in situ* sensors. Squares represents the main steps involved. Circles correspond to input and output.

2.1 Study region and data

To evaluate the effectiveness of our oddwater procedure we considered a challenging real-world problem of monitoring water-quality using *in situ* sensors in a natural river system. This is challenging because the system is susceptible to a wide range of environmental, biological and human impacts that can lead to variation in water-quality and affect the technological performance of the sensors. For comparison, we evaluated two study sites, Sandy Creek and Pioneer River (PR), both in the Mackay-Whitsunday region of northeastern Australia (Mitchell et al., 2005). These two rivers flow into the Great Barrier Reef lagoon, and have catchment areas of 1466 km^2 and 326 km^2, respectively. In this region, the wet season typically occurs from December to April and is dominated by higher rainfall and air temperatures, whereas the dry season typically occurs from May to November with lower rainfall and air temperatures (McInnes et al., 2015). The sensors at these two sites are housed within a monitoring station on the river banks. Water is pumped from the rivers to the stations approximately every 60 or 90 minutes to take measurements of various water-quality variables that are logged by the sensors. Here we focused on three water-quality variables: turbidity(NTU), conductivity (strictly, specific conductance at 25^0C; μS/cm) and river level (m).

The water-quality data obtained from *in situ* sensors located at Sandy Creek were available from 12 March 2017 to 12 March 2018. The data set included 5402 recorded points. These time series were irregular (i.e. the frequency of observations was not constant) with a minimum time gap of 10 minutes and a maximum time gap of around 4 hours. The data obtained from Pioneer River were available from 12 March 2017 to 12 March 2018, and included 6280 recorded points. Many missing values were observed during the initial part of all three series, turbidity, conductivity and river level, at Pioneer River. With the help of a group of water-quality experts, familiar with the study region and with over 40 years of combined knowledge of river water quality, observations were labeled as outliers or not, with the aim of evaluating the performance of the procedure. Our Shiny web application available through the *oddwater* R package was used during the labeling process to pinpoint observations and provide greater visual insight into the data. Using this interactive visualization tool and expert knowledge, the ground-truth labels were decided by consensus vote.

2.2 Apply rule-based approaches

Following Thottan and Ji (2003), we incorporated simple rules into our oddwater procedure to detect outliers such as out-of-range values, impossible values (e.g. negative values) and missing values, and labeled them prior to applying the statistical transformations introduced in Section 2.4.

If a sensor reading was outside the corresponding sensor detection range it was marked as an outlier. Negative readings are also inaccurate and impossible for river turbidity, conductivity and level. We therefore imposed a simple constraint on the algorithm to filter these values and mark them as outliers. Missing values are also frequently encountered in water-quality sensor data (Rangeti et al., 2015). We detected missing values by calculating the time gaps between readings. If a gap exceeded the maximum allowable time difference between any two consecutive readings, the corresponding time stamp was then marked as an outlier due to missingness. Here, the maximum allowable time difference was set at 180 minutes, given that the water-quality measurements were set to be taken at most every 90 minutes (measurements were often taken at higher frequencies during high-flow events, e.g. every 10-15 minutes, and occasionally as one-off measurements at times of interest to water managers).

2.3 Identify data features

After labeling out-of-range, impossible and missing values as outliers, further investigation was done with the remaining observations. We initiated this investigation by identifying common characteristics or patterns of the possible types of outliers in water-quality data that would differentiate them from typical instances or events. For turbidity, for example, "extreme" deviations upward are more likely than deviations downwards (Panguluri et al., 2009). The opposite is true for conductivity (Tutmez et al., 2006). Further, in a turbidity time series a sudden isolated upward shift (spike) is a point outlier (a single observation that is surprisingly large, independent of the neighboring observations (Goldstein & Uchida, 2016)), but if the sudden upward shift is followed by a gradually decaying tail then it becomes part of the typical behavior. For river level, rates of rise are often fast compared with fall rates. In general, isolated data points that are outside the general trend are outliers. Further, natural water processes under typical conditions generally tend to be comparatively slow; sudden changes therefore mostly correspond to outlying behaviors. Hereafter, these characteristics will be referred to as 'data features'.

2.4 Apply statistical transformations

After identifying the data features, different statistical transformations were applied to the time series to highlight different types of outliers, focusing on sudden isolated spikes, sudden isolated drops, sudden shifts, and clusters of spikes (Table 1) that deviate from the typical characteristics of each variable (Leigh et al., 2019).

In this work we considered the outlier detection problem in a multivariate setting. By applying different transformations on water-quality variables we converted our original problem of outlier detection in the temporal context to a non-temporal context through a high dimensional data space with three dimensions defined by the three variables: turbidity, conductivity and river level. Different transformations were applied on different axes of the three dimensional data space resulting in different data patterns. We evaluated the performance of the transformations (Dang & Wilkinson, 2014) using the maximum separability of the two classes: outliers and typical points in the three dimensional data space. For example, in the data obtained from Sandy Creek, the one sided derivative transformation clearly separated all of the target outlying points from the typical points, shown as either red triangles (corresponding to outliers) or green squares (corresponding to the immediate neighbours of outliers) (Figure 2(e)). To provide a better visual illustration, in Figure 2, we present only the two dimensional data space defined

Table 1. Transformation methods used to highlight different types of outliers in water-quality sensor data. Let Y_t represent an original series from one of the three variables: turbidity, conductivity and level at time t.

Transformation	**Formula**	**Data Feature**	**Focus**
Log transformation	$\log(y_t)$	High variability of the data.	To stabilize the variance across time series and make the patterns more visible (e.g. level shifts)
First difference	$\log(y_t/y_{t-1})$	Isolated spikes (in both positive and negative directions) that are outside the general trend are considered as outliers. Under typical behavior, sudden upward (downward) shifts are possible for turbidity (conductivity), but their rate of fall (rise) is generally slower than the rate of rise (fall).	To separate isolated spikes from the general upward/downward trend patterns.
Time gap	Δt		To identify missing values.
First derivative	$x_t = \log(y_t/y_{t-1})/\Delta t$	Data are unevenly spaced time series.	To handle irregular time series. Data points with large gaps will get small value. Large gaps indicate the lack of information to make a claim regarding the points.
One sided derivative			
Turbidity or level	$\min\{x_t, 0\}$	Extreme upward trend in turbidity and level under typical behavior.	To separate spikes from typical upward trends.
Conductivity	$\max\{x_t, 0\}$	Extreme downward trend in conductivity under typical behavior.	To separate isolated drops from typical downward trends.
Rate of change	$(y_t - y_{t-1})/y_t$	High or low variability in the data.	To detect change points in variance.
Relative difference	$y_t - (1/2)(y_{t-1} + y_{t+1})$	Natural processes are comparatively slow. Sudden changes (upward or downward movements) typically correspond to outlying instances.	To detect sudden changes (both upward and downward movements)

by turbidity and conductivity; however our actual data space is three dimensional. In this work our focus was to evaluate whether each point in time is an outlier or not, such that an alarm could be triggered in the presence of an outlier. However, it was not our interest to investigate which variable(s) is (are) responsible for the outlier in time. Therefore, in Figure 2, a point is marked as an outlier in the two dimensional space if at least one variable corresponding to that point was labelled as an outlier by the water-quality experts.

When the transformation involves both the current value Y_t and the lagged value Y_{t-1} (as in the first difference, first derivative, and one sided derivatives), the neighboring points can emerge as outliers instead of the actual outlying point. For an example, if an outlier occurs at time point t, then the two values derived from the first derivative transformation $((y_t - y_{t-1})$ and $(y_{t+1} - y_t))$ get highlighted as outlying values because they both involve y_t. That is each outlying instance is now represented by two consecutive values under the first derivative transformation. The goal of the one sided derivative transformation is to filter one high value for each outlying instance. However the high values obtained could correspond to either the actual outlying time point or the neighboring time point, because each transformed value is derived from two consecutive observations. If the primary focus of detecting technical outliers is to alert managers of sensor failures, then it will be inconsequential if the alarm is triggered either at the actual time point corresponding to the outlier or at the next immediate time point. However if the purpose is different, such as producing a trustworthy dataset by labeling or correcting detected outliers, then additional conditions should be imposed to ensure that the time points declared as outliers correspond to the actual outlying points and not to their immediate neighboring points.

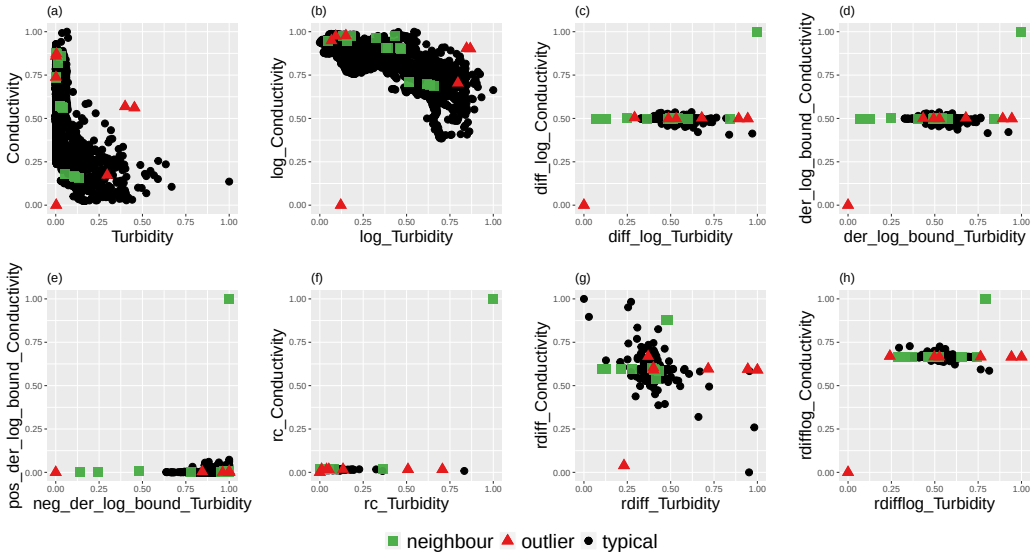

Figure 2. Bivariate relationships between transformed series of turbidity and conductivity measured by *in situ* sensors at Sandy Creek. In each scatter plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green. (a) Original series, (b) Log transformation, (c) First difference, (d) First derivative, (e) One sided derivative, and (f) Rate of change, (g) Relative difference (for original series), (h) Relative difference (for log transformed series). In each scatter plot, data are normalised such that they are bounded by the unit hypercube.

2.5 Calculate outlier scores

We considered eight unsupervised outlier scoring techniques for high dimensional data, involving nearest neighbor distances or densities of the observations and applied them to the three dimensional data space defined by the three variables: turbidity, conductivity and river level. Methods based on k-nearest neighbor distances (where $k \in Z^+$) were the NN-HD algorithm (details of this algorithm, which was inspired by HDoutliers algorithm (Wilkinson, 2018) are provided in Supporting Information), KNN-AGG and KNN-SUM algorithms (Angiulli & Pizzuti, 2002; Madsen, 2018) and Local Distance-based Outlier Factor (LDOF) algorithm (Zhang et al., 2009), which calculate the outlier score under the assumption that any outlying point (or outlying clusters of points) in the data space is (are) isolated; therefore the outliers are those points having the largest k-nearest neighbor distances. In contrast, the density based Local Outlier Factor (LOF) (Breunig et al., 2000), Connectivity-based Outlier Factor (COF) (Tang et al., 2002), Influenced Outlierness (INFLO) (Jin et al., 2006) and Robust Kernel-based Outlier Factor (RKOF) (Gao et al., 2011) algorithms calculate an outlier score based on how isolated a point is with respect to its surrounding neighbors, and therefore the outliers are those points having the lowest densities (see Supporting Information for detail). Each algorithm assigns outlier scores for all of the data points in the high dimensional space that described the degree of outlierness of the individual data points such that outliers are those points having the largest scores (Kriegel et al., 2010; Shahid et al., 2015). This step allowed us to set a data driven threshold (Section 2.6) for the outlier scores, to select the most relevant outliers (Chandola et al., 2009).

2.6 Calculate outlier threshold

Following Schwarz (2008), Burridge and Taylor (2006) and Wilkinson (2018), we used extreme value theory (EVT) to calculate a separate outlier threshold for each set of outlier scores calculated using a given unsupervised outlier scoring technique (introduced in Section 2.5) and assign a bivariate label for each point either as an outlier or typical point. Thus, 8 outlier scoring techniques resulted 8 different thresholds for a given dataset. The threshold calculation process started from a subset of data containing 50% of observations with the smallest outlier scores, under the assumption that this subset contained the outlier scores corresponding to typical data points and the remaining subset contained the scores corresponding to the possible candidates for outliers. Following Weissman's spacing theorem (Weissman, 1978), the algorithm then fit an exponential distribution to the upper tail of the outlier scores of the first subset, and computed the upper $1-\alpha$ (in this work α was set to 0.05) points of the fitted cumulative distribution function, thereby defining an outlying threshold for the next outlier score. From the remaining subset the algorithm then selected the point with the smallest outlier score. If this outlier score exceeded the cutoff point, all the points in the remaining subset were flagged as outliers and searching for outliers ceased. Otherwise, the point was declared as a non-outlier and was added to the subset of the typical points. The threshold was then updated by including the latest addition. The searching algorithm continued until an outlier score was found that exceeded the latest threshold (Schwarz, 2008). We performed this threshold calculation under the assumption that the distribution of outlier scores produced by each of the eight unsupervised outlier scoring techniques for high dimensional data was in the maximum domain of attraction of the Gumbel distribution which consists of distribution functions with exponentially decaying tails including the exponential, gamma, normal and log-normal (Embrechts et al., 2013).

2.7 Performance evaluation

In this paper, we focused on high priority outliers, as described in Leigh et al. (2019), in which importance ranking of different outlier types was done by taking into account the end-user goals and the potential impact of outliers going undetected. However, it is

beyond the scope of this paper to discuss in detail the different types of outliers and their importance ranking. For more detail, we refer the reader to Leigh et al. (2019). We performed an experimental evaluation on the accuracy and computational efficiency of our oddwater procedure with respect to the eight outlier scoring techniques, using the different transformations (Table 1) and different combinations of variables (turbidity, conductivity and river level). These experimental combinations were evaluated with respect to common measures for binary classification based on the values of the confusion matrix which summarizes the false positives (FP; i.e. when a typical observation is misclassified as an outlier), false negatives (FN; i.e. when an actual outlier is misclassified as a typical observation), true positives (TP; i.e. when an actual outlier is correctly classified), and true negatives (TN; i.e. when an observation is correctly classified as a typical point). In this work false positives and false negatives are equally undesirable as false positives may demand unnecessary and/or expensive actions for corrections and refinement and false negatives greatly reduce confidence in the data and results derived from them. The measures we considered include accuracy ($(TP+TN)/(TP+FP+FN+TN)$) which explains the overall effectiveness of a classifier; and geometric-mean (GM $= \sqrt{TP*TN}$) which explains the relative balance of TP and TN of the classifier (Sokolova & Lapalme, 2009). According to Hossin and Sulaiman (2015), these measures are not enough to capture the poor performance of the classifiers in the presence of imbalanced datasets where the size of the typical class (positive class) is much larger than the outlying class (negative class). The datasets obtained from *in situ* sensors were highly imbalanced and negatively dependent (i.e. containing many more typical observations than outliers). Therefore, we used three additional measures that are recommended for imbalanced problems with only two classes (i.e. typical and outlying) by Ranawana and Palade (2006): the negative predictive value ($NPV = TN/(FN + TN)$) which measures the probability of a negatively predicted pattern actually being negative; positive predictive value ($PPV = TP/(TP + FP)$) which measures the probability of a positively predicted pattern actually being positive; and optimized precision ($OP = P - RI$ where $P = S_pN_n + S_nN_p$; $RI = |S_p - S_n|/(S_p + S_n)$; $S_p = TN/(TN + FP)$; $S_n = TP/(TP+FN)$ and N_p and N_n represent the proportion of positives (outliers) and negatives (typical) within the entire dataset) which is a combination of accuracy, sensitivity and specificity metrics (Ranawana & Palade, 2006).

To evaluate the performance of our oddwater procedure we incorporated additional steps after detecting the outlying time points using the outlying threshold based on EVT. This was done because the time points declared as outliers by the outlying threshold could correspond to either the actual outlying points or to their neighbors. Once the time points were declared as outliers, the corresponding points in the three dimensional space were further investigated by comparing their positions with respect to the median of the typical points declared by the oddwater procedure. This step allowed us to find the most influential variable for each outlying point. For example, in Figure 2(e) the isolated point in the first quadrant is an outlier in the two dimensional space due to the outlying behavior of the conductivity measurement because the deviation of this point from the median happens primarily along the conductivity axis. In contrast, the four isolated points in the third quadrant are outliers due to the outlying behavior of the turbidity measurement because the deviations of the four points from the median happen primarily along the turbidity axis. After detecting the most influential variable for each outlying instance in the three dimensional space, further investigations were carried out separately for each individual outlying instance with respect to the most influential variable detected,to see whether the outlying instance was due to a sudden spike or a sudden drop by comparing the direction of the detected points with respect to the mean of its two immediate surrounding neighbors and itself. These additional steps in the oddwater procedure allowed us to trigger an alarm at the actual outlying point in time if the neighboring points were declared as outliers instead of the actual outliers. However, we acknowledge that these additional steps select only the most influential variable, not all of the influential variables in the presence of more than one influential variable. The additional steps were

incorporated solely to measure the performance of the oddwater procedure. In practice, and because the goal is to trigger an alarm in an occurrence of a technical outlier, it is inconsequential if the alarm is triggered either at the actual time point or at the immediate neighbouring time points corresponding to the actual outlier. As such, users of the oddwater procedure can ignore these additional steps.

Using the outlier threshold, our oddwater procedure assigns a bivariate label (either as outlier or typical point) to each observed time point and thereby creates a vector of predicted class labels. That is, if a time point is declared as an outlier by oddwater procedure, then that could be due to at least one variable in the dataset. We also declared each time point as an outlier or not based on the labels assigned by the water quality experts. At a given time point, if at least one variable was labeled as an outlier by the water quality experts then the corresponding time point was marked as an outlier and thereby creating a vector of ground-truth labels. Then the performance measures were calculated based on these two vectors of ground-truth labels and predicted class labels. Thus, this performance evaluation was done with respect to the algorithm's ability to label a point in time as an outlier or not (i.e., a point in time is an outlier if the observed value for any one or more of the three variables measured at that point in time are outliers).

2.8 Software implementation

The oddwater procedure was implemented in the open source R package `oddwater` (Talagala & Hyndman, 2018), which provides a growing list of transformation and outlier scoring methods for high dimensional data together with visualization and performance evaluation techniques. Version 0.6.0 of the package `oddwater` was used for the results presented herein and is available from Github (https://github.com/pridiltal/oddwater). In addition to the implementations available through `oddwater` package, `DDoutlier` package (Madsen, 2018) was also used for outlier score calculations. We measured the computation time (minimum (min_t), mean (mu_t), maximum (max_t) execution time) using the `microbenchmark` package (Mersmann, 2018) for different combinations of algorithms, transformations and variable combinations on 28 core Xeon-E5-2680-v4 @ 2.40GHz servers. We also developed an R Shiny application (available via `oddwater` R package) to provide interactive visual analytic tools to gain greater insight into the data and perform preliminary investigations of the relationships between water-quality variables at different sites. Further, we have archived a snapshot of version 0.6.0 of the R package on Zenodo (DOI: 10.5281/zenodo.2890469) along with the code and datasets used, with the aim of facilitating reproducibility of the results presented herein.

3 Results

3.1 Analysis of water-quality data from *in situ* sensors at Sandy Creek

A negative relationship was clearly visible between the water-quality variables: turbidity and conductivity and also between conductivity and river level measured by *in situ* sensors at Sandy Creek (Figures 3 and 4(a,c)). Further, no clear separation was observed between the target outliers and the typical points in the original data space (Figure 4(a–c)). However, a clear separation was apparent between the two sets of points once the one sided derivative transformation (an appropriate transformation for unevenly spaced data) was applied to the original series (Figures 4(d–f) and 5).

KNN-AGG and KNN-SUM algorithms performed on all three water-quality variables together, and on turbidity and conductivity together using the one sided derivative transformation, gave the highest OP (0.8329) and NPV values(0.9996), which are the most recommended measurements for negatively dependent data where the focus is

more on sensitivity (the proportion of positive patterns being correctly recognized as being positive) than specificity (Ranawana & Palade, 2006).

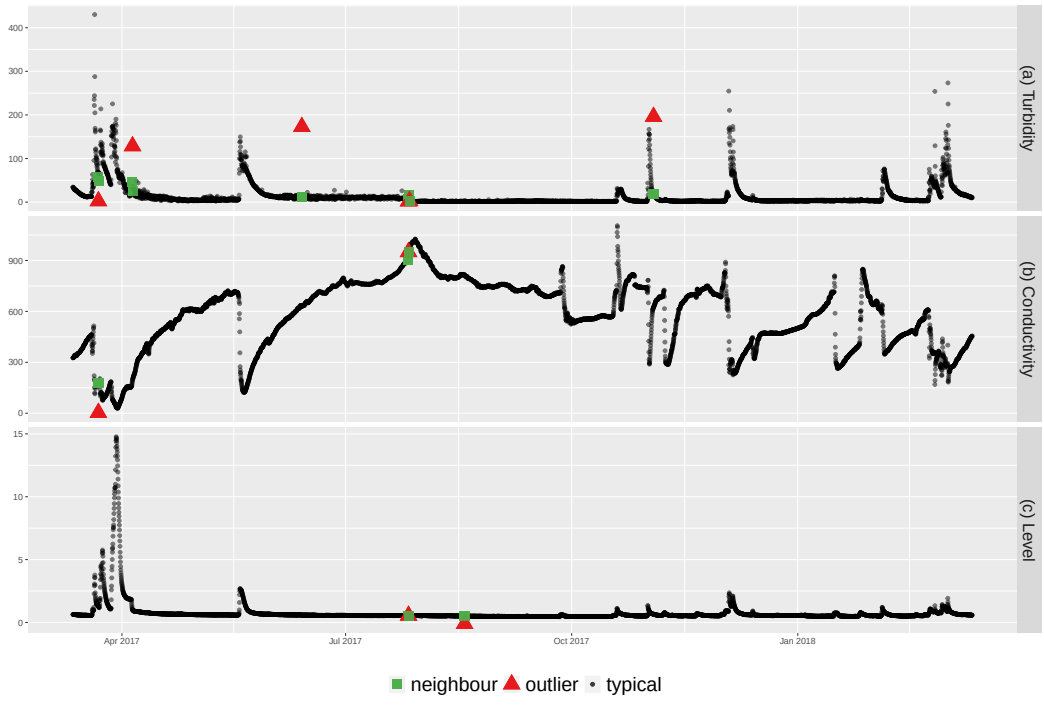

Figure 3. Time series for turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Sandy Creek. In each plot, outliers determined by water-quality experts are shown in red. Typical points are shown in black.

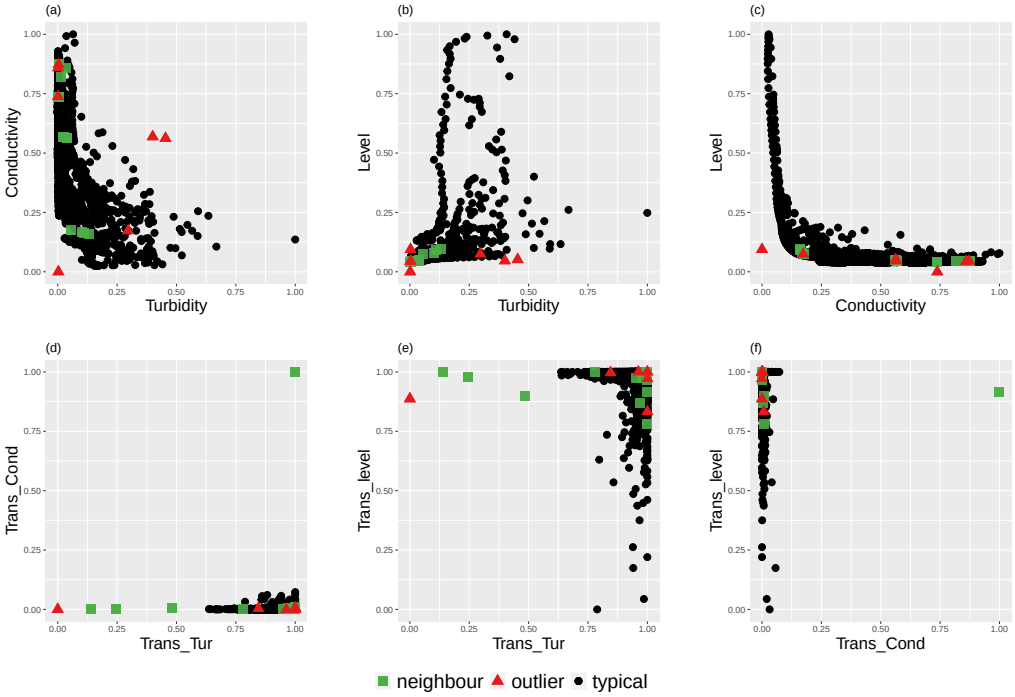

Figure 4. Top panel (a–c): Bi-variate relationships between original water-quality variables (turbidity (NTU), conductivity (μS/cm) and river level (m)) measured by *in situ* sensors at Sandy Creek. Bottom panel (d–f): Bi-variate relationships between transformed series (one sided derivative) of turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Sandy Creek. In each scatter plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green.

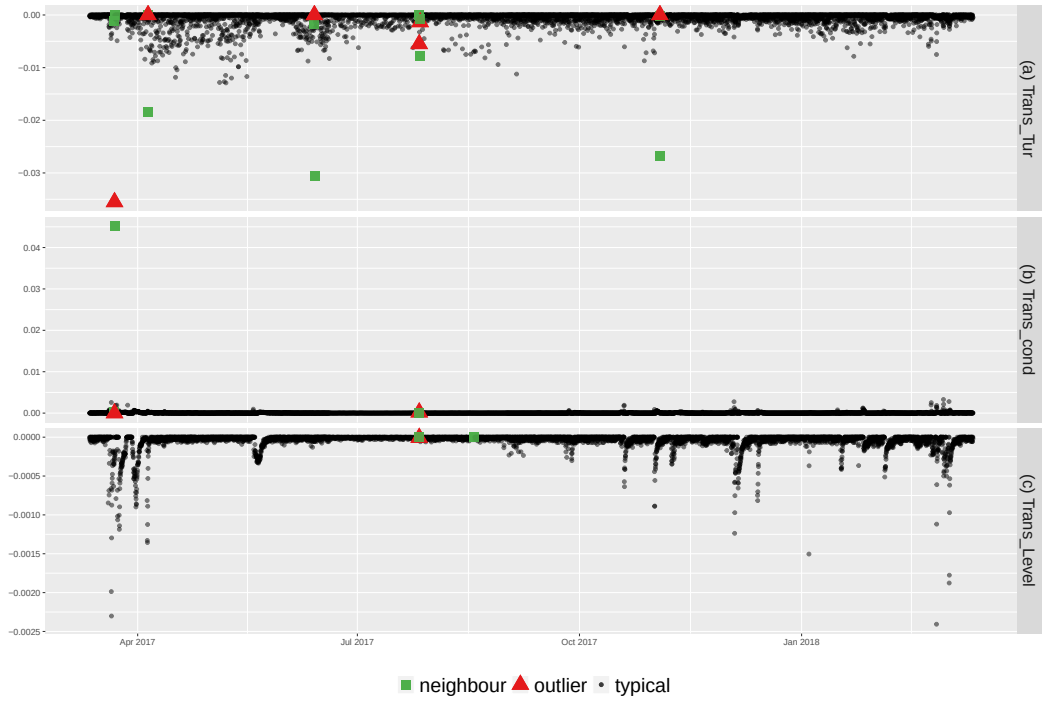

Figure 5. Transformed series (one sided derivatives) of turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Sandy Creek. In each plot outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green

Based on OP values, the one sided derivative transformation outperformed the first derivative transformation (Table 2, rows 1–5 compared to rows 6–10). Further, the distance-based outlier detection algorithms NN-HD , KNN-AGG and KNN-SUM outperformed all others (Table 2, rows 1–10 compared to rows 11–48). Among the three methods the performance of k-nearest neighbor distance-based algorithms were only slightly higher (OP = 0.8329) than the NN-HD algorithm (OP= 0.7996), which is based only on the nearest neighbor distance. The algorithm combinations with the five highest OP values also had high PPV (approximately 0.8). Furthermore, considering river level for the detection of outliers in the water-quality sensors slightly improved the performance (OP = 0.8329). Among the analysis with transformed series, LOF with the first derivative transformation performed the least well (OP= 0.2489). For most of the outlier detection algorithms (KNN-SUM, KNN-AGG, NN-HD , COF, LOF and INFLO) the poorest performances were associated with the untransformed original series, having the lowest OP and NPV values, highlighting how data transformation can improve the ability of outlier detection algorithms while maintaining low false detection rates.

Table 2. Performance metrics of outlier detection algorithms performed on multivariate water-quality time series data (T, turbidity; C, conductivity; L, river level) from in situ sensors at Sandy Creek, arranged in descending order of OP values. See Sections 2.7-8 for performance metric codes and details.

i	Variables	Transformation	Method	TN	FN	FP	TP	Accuracy	GM	OP	PPV	NPV	min_t	mu_t	max_t
1	T-C-L	One sided Derivative	KNN-AGG	5394	2	1	5	0.9994	164.23	0.83	0.83	0.9996	378.5	404.0	493.4
2	T-C-L	One sided Derivative	KNN-SUM	5394	2	1	5	0.9994	164.23	0.83	0.83	0.9996	177.2	186.8	270.3
3	T-C	First Derivative	NN-HD	5393	2	3	4	0.9991	146.87	0.80	0.57	0.9996	40.5	45.0	72.1
4	T-C	First Derivative	KNN-AGG	5392	2	4	4	0.9989	146.86	0.80	0.50	0.9996	386.3	415.8	489.4
5	T-C	One sided Derivative	NN-HD	5396	2	0	4	0.9996	146.91	0.80	1.00	0.9996	102.7	112.9	195.4
6	T-C	One sided Derivative	KNN-AGG	5395	2	1	4	0.9994	146.90	0.80	0.80	0.9996	381.8	411.7	518.0
7	T-C	One sided Derivative	KNN-SUM	5395	2	1	4	0.9994	146.90	0.80	0.80	0.9996	177.9	190.4	286.2
8	T-C-L	First Derivative	KNN-AGG	5395	4	0	3	0.9993	127.22	0.60	1.00	0.9993	377.6	404.4	476.2
9	T-C-L	First Derivative	KNN-SUM	5395	4	0	3	0.9993	127.22	0.60	1.00	0.9993	179.2	188.9	273.3
10	T-C	First Derivative	KNN-SUM	5396	4	0	2	0.9993	103.88	0.50	1.00	0.9993	179.0	189.5	283.5
11	T-C	First Derivative	LDOF	5395	4	1	2	0.9991	103.87	0.50	0.67	0.9993	17261.5	17444.7	17809.9
12	T-C	One sided Derivative	LDOF	5395	4	1	2	0.9991	103.87	0.50	0.67	0.9993	17024.3	17253.8	18079.4
13	T-C-L	First Derivative	NN-HD	5395	5	0	2	0.9991	103.87	0.44	1.00	0.9991	48.7	52.5	66.9
14	T-C-L	First Derivative	INFLO	5381	5	14	2	0.9965	103.74	0.44	0.12	0.9991	1076.5	1107.9	1168.3
15	T-C-L	First Derivative	COF	5393	5	2	2	0.9987	103.86	0.44	0.50	0.9991	5869.1	5939.8	6394.2
16	T-C-L	First Derivative	RKOF	5380	5	15	2	0.9963	103.73	0.44	0.12	0.9991	341.6	369.7	456.1
17	T-C-L	One sided Derivative	NN-HD	5395	5	0	2	0.9991	103.87	0.44	1.00	0.9991	110.4	118.2	193.2
18	T-C-L	One sided Derivative	INFLO	5392	5	3	2	0.9985	103.85	0.44	0.40	0.9991	1071.5	1113.6	1177.6
19	T-C-L	One sided Derivative	COF	5393	5	2	2	0.9987	103.86	0.44	0.50	0.9991	5676.8	5787.4	6238.4
20	T-C-L	One sided Derivative	LDOF	5392	5	3	2	0.9985	103.85	0.44	0.40	0.9991	17181.5	17261.9	17435.8
21	T-C-L	One sided Derivative	LOF	5392	5	3	2	0.9985	103.85	0.44	0.40	0.9991	500.2	516.9	596.9
22	T-C-L	One sided Derivative	RKOF	5387	5	8	2	0.9976	103.80	0.44	0.20	0.9991	338.9	370.5	464.0
23	T-C-L	Original series	KNN-AGG	5394	5	1	2	0.9989	103.87	0.44	0.67	0.9991	376.6	391.6	465.3
24	T-C-L	Original series	INFLO	5386	5	9	2	0.9974	103.79	0.44	0.18	0.9991	1034.3	1070.7	1136.7
25	T-C-L	Original series	LDOF	5393	5	2	2	0.9987	103.86	0.44	0.50	0.9991	17078.5	17156.9	17308.1
26	T-C-L	Original series	RKOF	5392	5	3	2	0.9985	103.85	0.44	0.40	0.9991	322.3	354.0	426.4
27	T-C	First Derivative	INFLO	5392	5	4	1	0.9983	73.43	0.28	0.20	0.9991	1134.6	1194.9	1271.1
28	T-C	First Derivative	COF	5396	5	0	1	0.9991	73.46	0.28	1.00	0.9991	5881.5	5991.8	6552.4
29	T-C	First Derivative	LOF	5394	5	2	1	0.9987	73.44	0.28	0.33	0.9991	498.3	512.3	596.1
30	T-C	First Derivative	RKOF	5392	5	4	1	0.9983	73.43	0.28	0.20	0.9991	335.1	363.2	435.7
31	T-C	One sided Derivative	INFLO	5394	5	2	1	0.9987	73.44	0.28	0.33	0.9991	1153.1	1207.0	1281.9
32	T-C	One sided Derivative	COF	5394	5	2	1	0.9987	73.44	0.28	0.33	0.9991	5755.0	5880.8	6420.8
33	T-C	One sided Derivative	LOF	5384	5	12	1	0.9969	73.38	0.28	0.08	0.9991	501.1	511.3	585.7
34	T-C	One sided Derivative	RKOF	5380	5	16	1	0.9961	73.35	0.28	0.06	0.9991	339.5	368.3	456.6
35	T-C	Original series	KNN-AGG	5395	5	1	1	0.9989	73.45	0.28	0.50	0.9991	371.5	405.1	483.7
36	T-C	Original series	INFLO	5387	5	9	1	0.9974	73.40	0.28	0.10	0.9991	1095.2	1143.6	1219.4
37	T-C	Original series	LDOF	5394	5	2	1	0.9987	73.44	0.28	0.33	0.9991	16842.1	17022.9	17414.4
38	T-C	Original series	RKOF	5393	5	3	1	0.9985	73.44	0.28	0.25	0.9991	321.0	351.8	440.3
39	T-C-L	First Derivative	LDOF	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	17253.9	17323.2	17400.9
40	T-C-L	First Derivative	LOF	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	504.6	517.1	604.4
41	T-C-L	Original series	NN-HD	5394	6	1	1	0.9987	73.44	0.25	0.50	0.9989	45.4	48.6	60.3
42	T-C-L	Original series	KNN-SUM	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	164.7	177.3	243.4
43	T-C-L	Original series	COF	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	5864.4	5931.7	6329.7
44	T-C-L	Original series	LOF	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	480.2	505.0	576.2
45	T-C	Original series	NN-HD	5395	6	1	0	0.9987	0.00	0.00	0.00	0.9989	38.1	41.7	66.3
46	T-C	Original series	KNN-SUM	5396	6	0	0	0.9989	0.00	0.00	NaN	0.9989	172.7	184.6	272.5
47	T-C	Original series	COF	5396	6	0	0	0.9989	0.00	0.00	NaN	0.9989	5826.3	5896.4	6804.3
48	T-C	Original series	LOF	5396	6	0	0	0.9989	0.00	0.00	NaN	0.9989	477.0	502.7	568.0

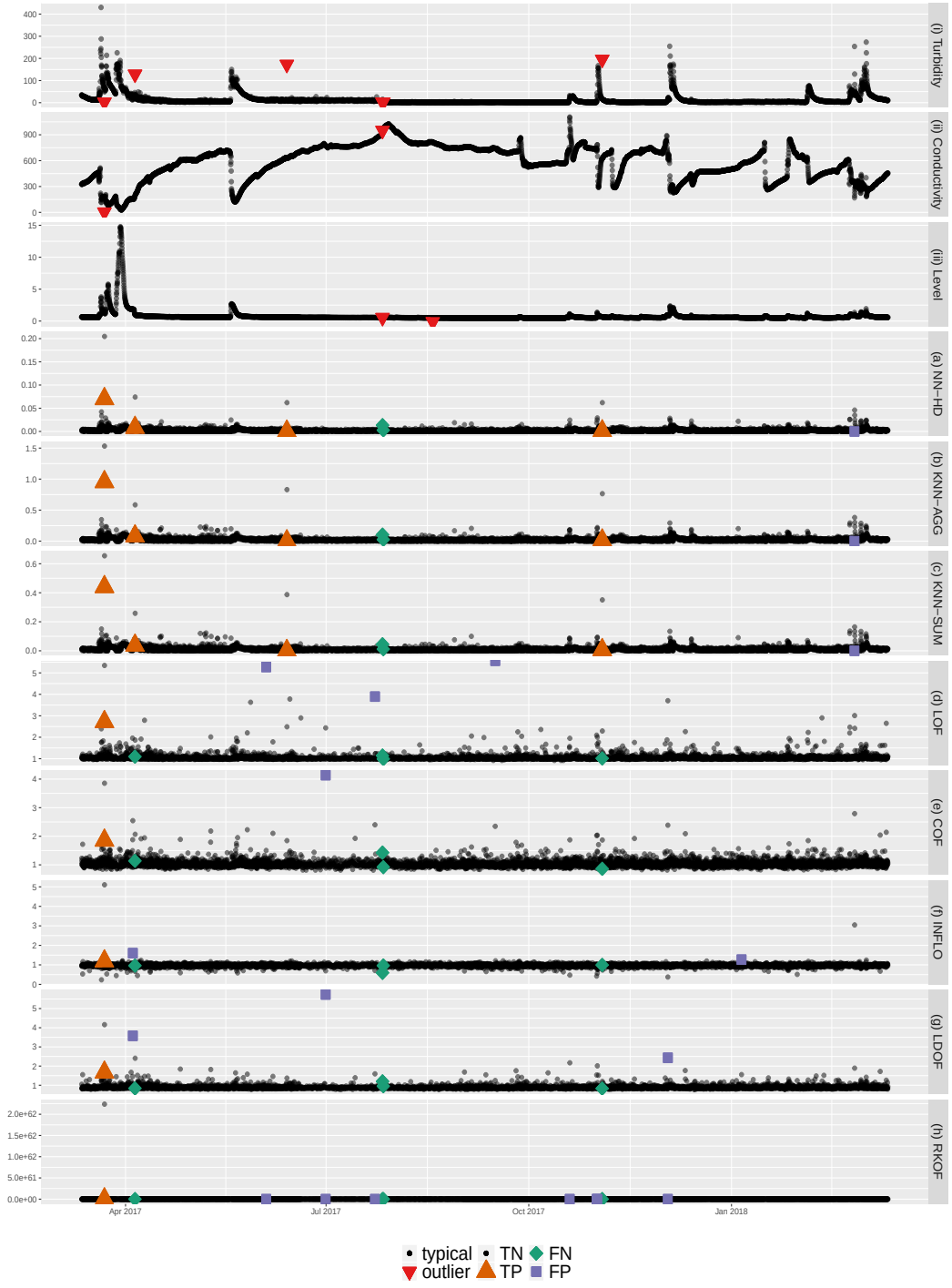

Figure 6. Classification of outlier scores produced from different algorithms as true negatives (TN), true positives (TP), false negatives (FN), false positives (FP). The top three panels (i, ii, iii) correspond to the original series (turbidity, conductivity and river level) measured by *in situ* sensors at Sandy Creek. The target outliers (detected by water-quality experts) are shown in red, while typical points are shown in black. The remaining panels (a–h) give outlier scores produced by different outlier detection algorithms for high dimensional data when applied to the transformed series (one sided derivative) of the three variables: turbidity, conductivity and level. Through different outlier scoring algorithms (Panel a - h) we are evaluating whether each point in time is an outlier or not. Therefore, from Panel a-h, if the outlier scoring algorithm is effective, then there should be either TP or TN at each point in time when either a red triangle is plotted in at least one of the three panels (i- iii), or black dots are plotted in all of the top three panels (i - iii), respectively. Since outlier scores are non negative and are mostly clustered near zero, with some occasional high values, a square root transformation was applied to reduce skewness of the data in Panel (a) to (h).

The three outlier detection algorithms that demonstrated the highest level of accuracy (NN-HD, KNN-AGG and KNN-SUM) also outperformed the others with respect to computational time. NN-HD algorithm required the least computational time. Among the remaining two, the mean computational time of KNN-AGG ($\approx$ 400 milliseconds) was twice that of KNN-SUM's ($<$ 200 milliseconds). LOF and its extensions (INFLO, COF and LDOF) demonstrated the poorest performance with respect computational time ($>$ 500 milliseconds on average).

Only KNN-SUM and KNN-AGG assigned high scores to most of the targeted outliers in turbidity, conductivity and level data transformed using the one-sided derivative (Figure 6(a,b)). For each outlying instance however, the next immediate neighboring point was assigned the high outlier score instead of the true outlying point. After determining the most influential variable using the additional steps of the algorithm (Section 2.7), adjustments were made to correct this to the actual outlier. The outlier scores produced by LOF and COF (Figure 6(d,f)) were unable to capture the outlying behaviors correctly and demonstrated high scattering. In comparison to other outlier scoring algorithms, KNN-SUM algorithm displayed a good compromise between accuracy and computational efficiency (Table 2).

3.2 Analysis of water-quality data from *in situ* sensors at Pioneer River

Some of the target outliers in the data obtained from the *in situ* sensors at Pioneer River only deviated slightly from the general trend (Figure 7), making outlier detection challenging. A negative relationship was clearly visible between turbidity and conductivity (Figure 8(a)), however the relationship between level and conductivity was complex (Figure 8(c)). Most of the target outliers were masked by the typical points in the original space (Figure 8(a–c)). Similar to Sandy Creek, data obtained from the sensors at Pioneer River showed good separation between outliers and typical points under the one sided derivative transformation (Figures 8(d–f) and 9). However, the sudden spikes in turbidity labeled as outliers by water-quality experts could not be separated from the majority by a large distance and were only visible as a small group (micro cluster (Goldstein & Uchida, 2016)) in the boundary defined by the typical points (Figure 8(d, e)).

From the performance analysis it was observed that turbidity and conductivity together produced better results (Table 3, rows 1–3) than when combined with river level, which tended to reduce the performance (i.e. generating lower OP and NPV values) while increasing the false negative rate (Table 3, rows 4–5). KNN-AGG and KNN-SUM (Table 3, rows 1–3) had the highest accuracy (0.9978), lowest error rates (0.0022), highest geometric means (492.8012), highest OP (0.8845) and highest NPV (0.9984). Despite the challenge given by the small spikes which could not be clearly separated from the typical points, KNN-AGG, KNN-SUM and NN-HD with one sided derivatives of turbidity and conductivity still detected some of those points as outliers while maintaining low false negative and false positive rates. Similar to Sandy Creek, NN-HD ($<$ 200 milliseconds on average) and KNN-SUM ($<$ 230 milliseconds on average) demonstrated the highest computational efficiency for the data obtained from Pioneer River.

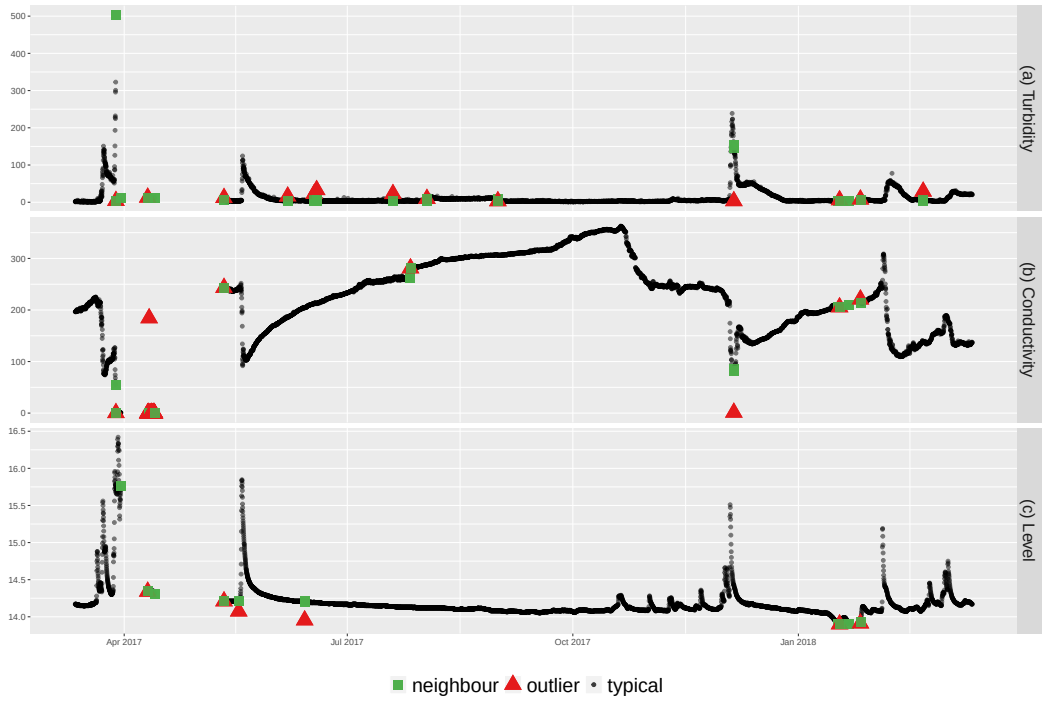

Figure 7. Time series for turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Pioneer River. In each plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black.

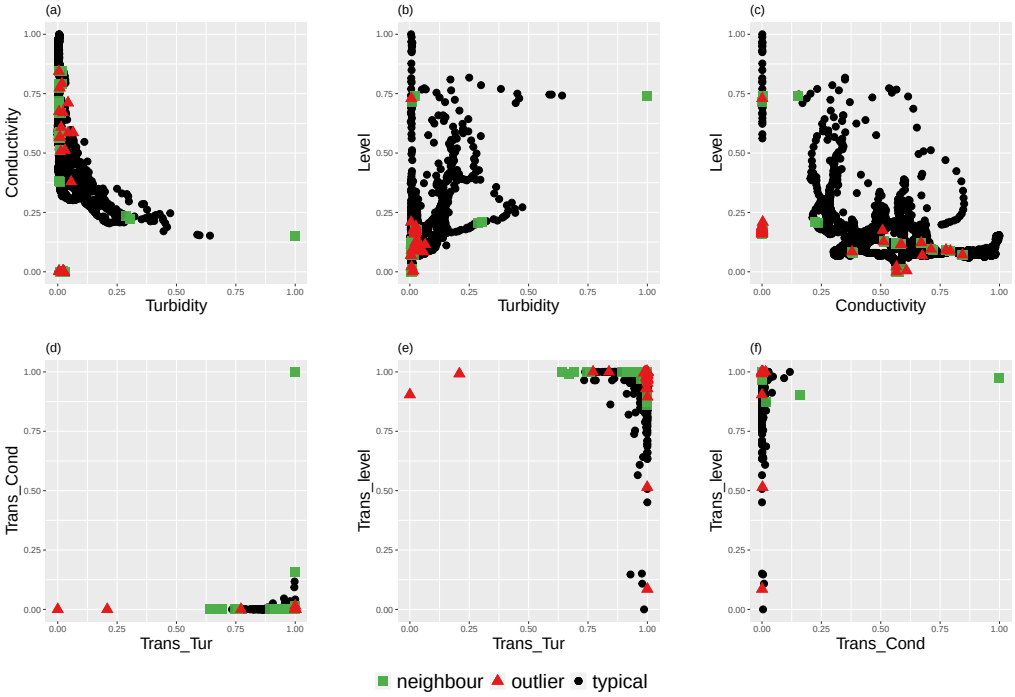

Figure 8. Top panel (a–c): Bi-variate relationships between original water-quality variables (turbidity (NTU), conductivity (μS/cm) and river level (m)) measured by *in situ* sensors at Pioneer River. Bottom panel (d–f): Bi-variate relationships between transformed series (one sided derivative) of turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Pioneer River. In each scatter plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green.

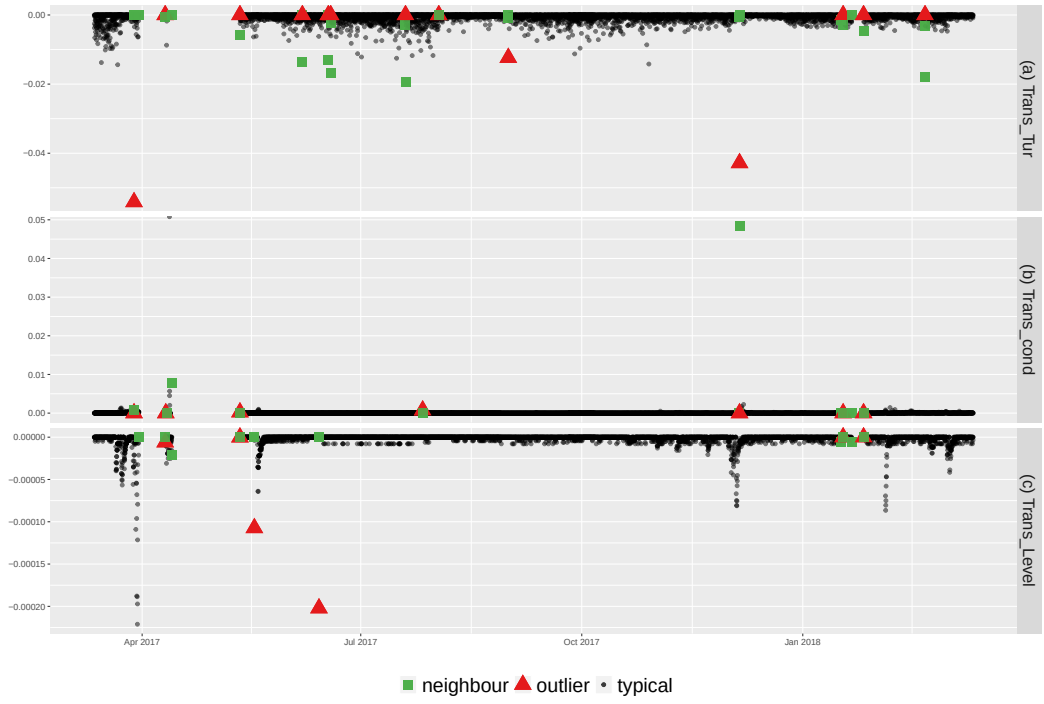

Figure 9. Transformed series (one sided derivatives) of turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Pioneer River. In each plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green

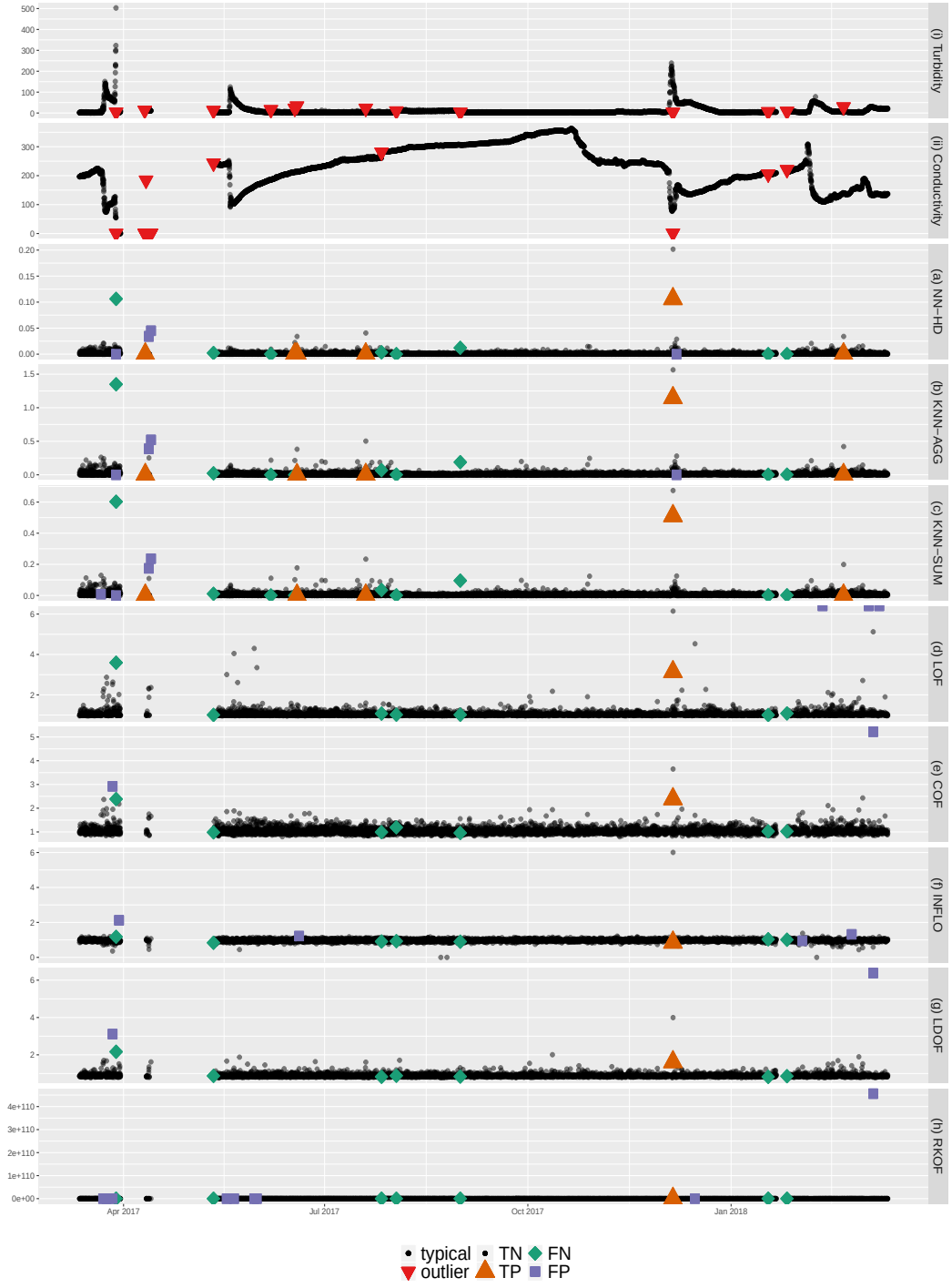

Figure 10. Classification of outlier scores produced from different algorithms as true negatives (TN), true positives (TP), false negatives (FN), false positives (FP). The top two panels (i and ii) correspond to the original series (turbidity and conductivity) measured by *in situ* sensors at Pioneer River. The target outliers (detected by water-quality experts) are shown in red, while typical points are shown in black. The remaining panels (a–h) give outlier scores produced by different outlier detection algorithms for high dimensional data when applied to the transformed series (one sided derivative) of the two variables: turbidity and conductivity. Through different outlier scoring algorithms (Panel a - h) we are evaluating whether each point in time is an outlier or not. Therefore, from Panel a-h, if the outlier scoring algorithm is effective, then there should be either TP or TN at each point in time when either a red triangle is plotted in at least one of the two panels (i- ii), or black dots are plotted in both of the top two panels (i - ii), respectively. Since outlier scores are non negative and are mostly clustered near zero, with some occasional high values, a square root transformation was applied to reduce skewness of the data in Panel (a) to (h).

Table 3. Performance metrics of outlier detection algorithms performed on multivariate water-quality time series data (T, turbidity; C, conductivity; L, river level) from in situ sensors at Pioneer River, arranged in descending order of OP values. See Sections 2.7-8 for performance metric codes and details.

i	Variables	Transformation	Method	TN	FN	FP	TP	Accuracy	GM	OP	PPV	NPV	min_t	mu_t	max_t
1	T-C	One sided Derivative	NN-HD	6226	10	5	39	0.9976	492.76	0.88	0.89	0.9984	128.0	136.5	257.7
2	T-C	One sided Derivative	KNN-AGG	6227	10	4	39	0.9978	492.80	0.88	0.91	0.9984	443.5	478.8	564.6
3	T-C	One sided Derivative	KNN-SUM	6227	10	4	39	0.9978	492.80	0.88	0.91	0.9984	209.6	222.2	325.5
4	T-C	First Derivative	NN-HD	6229	12	2	37	0.9978	480.08	0.86	0.95	0.9981	169.6	182.0	272.3
5	T-C	First Derivative	KNN-AGG	6229	12	2	37	0.9978	480.08	0.86	0.95	0.9981	449.5	488.5	588.2
6	T-C	First Derivative	KNN-SUM	6229	12	2	37	0.9978	480.08	0.86	0.95	0.9981	212.1	225.3	325.9
7	T-C	First Derivative	INFLO	6225	12	6	37	0.9971	479.92	0.86	0.86	0.9981	1452.1	1525.0	1613.2
8	T-C	First Derivative	RKOF	6224	12	7	37	0.9970	479.88	0.86	0.84	0.9981	400.2	430.4	523.9
9	T-C-L	One sided Derivative	KNN-AGG	6225	12	4	39	0.9975	492.72	0.86	0.91	0.9981	437.4	465.2	541.6
10	T-C-L	One sided Derivative	KNN-SUM	6225	12	4	39	0.9975	492.72	0.86	0.91	0.9981	195.6	214.5	297.8
11	T-C-L	First Derivative	RKOF	6211	13	18	38	0.9951	485.82	0.85	0.68	0.9979	396.9	425.9	503.4
12	T-C-L	First Derivative	KNN-AGG	6227	14	2	37	0.9975	480.00	0.84	0.95	0.9978	460.8	478.0	570.3
13	T-C-L	First Derivative	KNN-SUM	6227	14	2	37	0.9975	480.00	0.84	0.95	0.9978	201.5	220.0	292.2
14	T-C	First Derivative	COF	6230	13	1	36	0.9978	473.58	0.84	0.97	0.9979	7812.7	7908.2	8453.3
15	T-C	First Derivative	LDOF	6230	13	1	36	0.9978	473.58	0.84	0.97	0.9979	23241.0	23435.7	24522.1
16	T-C	First Derivative	LOF	6228	13	3	36	0.9975	473.51	0.84	0.92	0.9979	562.6	594.4	668.3
17	T-C	One sided Derivative	INFLO	6227	13	4	36	0.9973	473.47	0.84	0.90	0.9979	1488.8	1559.9	1633.1
18	T-C	One sided Derivative	COF	6229	13	2	36	0.9976	473.54	0.84	0.95	0.9979	7393.6	7505.5	8037.1
19	T-C	One sided Derivative	LDOF	6228	13	3	36	0.9975	473.51	0.84	0.92	0.9979	22802.2	22986.0	23561.3
20	T-C	One sided Derivative	LOF	6228	13	3	36	0.9975	473.51	0.84	0.92	0.9979	581.9	596.9	682.6
21	T-C	One sided Derivative	RKOF	6219	13	12	36	0.9960	473.16	0.84	0.75	0.9979	388.6	419.7	510.4
22	T-C	Original Series	INFLO	6227	13	4	36	0.9973	473.47	0.84	0.90	0.9979	1405.6	1498.5	1578.2
23	T-C-L	First Derivative	COF	6228	15	1	36	0.9975	473.51	0.83	0.97	0.9976	7823.0	7910.7	8344.7
24	T-C-L	First Derivative	LDOF	6228	15	1	36	0.9975	473.51	0.83	0.97	0.9976	23220.1	23357.7	23878.3
25	T-C-L	One sided Derivative	NN-HD	6228	15	1	36	0.9975	473.51	0.83	0.97	0.9976	125.7	131.9	206.1
26	T-C	Original Series	NN-HD	6230	14	1	35	0.9976	466.96	0.83	0.97	0.9978	159.9	171.0	278.2
27	T-C	Original Series	KNN-AGG	6226	14	5	35	0.9970	466.81	0.83	0.88	0.9978	434.2	468.7	553.0
28	T-C	Original Series	KNN-SUM	6226	14	5	35	0.9970	466.81	0.83	0.88	0.9978	192.9	211.6	305.8
29	T-C	Original Series	COF	6231	14	0	35	0.9978	467.00	0.83	1.00	0.9978	7518.5	7617.6	8501.1
30	T-C	Original Series	LDOF	6231	14	0	35	0.9978	467.00	0.83	1.00	0.9978	22770.9	22910.4	23857.1
31	T-C	Original Series	LOF	6231	14	0	35	0.9978	467.00	0.83	1.00	0.9978	551.2	579.1	632.6
32	T-C	Original Series	RKOF	6222	14	9	35	0.9963	466.66	0.83	0.80	0.9978	373.6	401.9	475.4
33	T-C-L	First Derivative	NN-HD	6227	15	2	36	0.9973	473.47	0.82	0.95	0.9976	157.3	167.1	244.3
34	T-C-L	One sided Derivative	INFLO	6226	15	3	36	0.9971	473.43	0.82	0.92	0.9976	1383.5	1418.8	1477.4
35	T-C-L	One sided Derivative	COF	6227	15	2	36	0.9973	473.47	0.82	0.95	0.9976	7414.6	7497.9	7899.6
36	T-C-L	One sided Derivative	LDOF	6227	15	2	36	0.9973	473.47	0.82	0.95	0.9976	22756.8	23090.7	23941.1
37	T-C-L	One sided Derivative	RKOF	6214	15	15	36	0.9952	472.97	0.82	0.71	0.9976	390.5	422.1	490.3
38	T-C-L	First Derivative	INFLO	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	1344.7	1398.3	1456.9
39	T-C-L	First Derivative	LOF	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	585.6	600.7	688.1
40	T-C-L	One sided Derivative	LOF	6223	16	6	35	0.9965	466.70	0.81	0.85	0.9974	583.4	596.1	672.1
41	T-C-L	Original Series	NN-HD	6228	16	1	35	0.9973	466.88	0.81	0.97	0.9974	152.8	163.0	231.2
42	T-C-L	Original Series	KNN-AGG	6224	16	5	35	0.9967	466.73	0.81	0.88	0.9974	439.3	456.3	534.2
43	T-C-L	Original Series	KNN-SUM	6224	16	5	35	0.9967	466.73	0.81	0.88	0.9974	186.5	201.4	269.6
44	T-C-L	Original Series	INFLO	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	1329.9	1372.8	1415.0
45	T-C-L	Original Series	COF	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	7596.5	7707.2	8357.8
46	T-C-L	Original Series	LDOF	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	22897.7	127337.1	10458496.0
47	T-C-L	Original Series	LOF	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	549.5	580.9	646.9
48	T-C-L	Original Series	RKOF	6217	16	12	35	0.9955	466.47	0.81	0.74	0.9974	368.3	406.8	497.2

4 Discussion

We introduced a new procedure, named oddwater procedure for the detection of outliers in water-quality data from *in situ* sensors, where outliers were specifically defined as due to technical errors that make the data unreliable and untrustworthy. We showed that our oddwater procedure, with carefully selected data transformation methods derived from data features, can greatly assist in increasing the performance of a range of existing outlier detection algorithms. Our oddwater procedure, and analysis using data obtained from *in situ* sensors positioned at two study sites, Sandy Creek and Pioneer River, performed well with outlier types such as sudden isolated spikes, sudden isolated drops, and level shifts, while maintaining low false detection rates. As an unsupervised procedure, our approach can be easily extended to other water-quality variables, other sites and also to other outlier detection tasks in other application domains. The only requirement is to select suitable transformation methods according to the data features that differentiate the outlying instances from the typical behaviors of a given system.

Studies have shown that transforming variables affects densities, relative distances and orientation of points within the data space and therefore can improve the ability to perceive patterns in the data which are not clearly visible in the original data space (Dang & Wilkinson, 2014). This was the case in our study, where no clear separation was visible between outliers and typical data points in the original data space, but a clear separation was obtained between the two sets of points once the one-sided derivative transformation was applied to the original series. Having this type of a separation between outliers and typical points is important before applying unsupervised outlier detection algorithms for high dimensional data because the methods are usually based on the definition of outliers in terms of distance or density (Talagala et al., 2018). Most of the outlier detection algorithms (KNN-SUM, KNN-AGG, NN-HD , COF, LOF and INFLO) performed least well with the untransformed original series, demonstrating how data transformation methods can assist in improving the ability of outlier detection algorithms while maintaining low false detection rates.

Although outlying points were clearly separated from their majority, which corresponded to the typical behaviors, the individual outliers were not isolated and were surrounded by the other outlying points. Because NN-HD has the additional requirement of isolation in addition to clear separation between outlying points and typical points, it performed poorly in comparison to the two KNN distance-based algorithms (KNN-AGG and KNN-SUM) which are not restricted to the single most nearest neighbor (Talagala et al., 2018). For the current work k was set to 10, the maximum default value of k in Madsen (2018), because too large a value of k could skew the focus towards global outliers (points that deviates significantly from the rest of the dataset) alone (Zhang et al., 2009) and make the algorithms computationally inefficient. On the other hand, too small a value of k could incorporate an additional assumption of isolation into the algorithm, as in the NN-HD algorithm where $k = 1$. Among the analysis using transformed series, LOF with the first derivative transformation performed the least well, which could also be due to its additional assumption of isolation (Tang et al., 2002). However, using the same k across all algorithms may bias direct comparison as the performance of the algorithms can depend on the value of k and algorithms can reach their peak performance for different choices of k (Campos et al., 2016). Therefore performing an optimisation to select the best k is non trivial and we leave it for future work.

We took the correlation structure between the variables into account when detecting outliers as some were apparent only in the high dimensional space but not when each variable was considered independently (Ben-Gal, 2005). A negative relationship was observed between conductivity and turbidity and also between conductivity and level for the Sandy Creek data. However, for Pioneer River, no clear relationship was observed between level and the remaining two variables, turbidity and conductivity. This could be one reason why the variable combination with river level gave poor results for the Pioneer River dataset, while results for other combinations were similar to those of Sandy Creek.

The one-sided derivative transformation outperformed the derivative transformation. This was expected because in an occurrence of a sudden spike or isolated drop the first derivative assigns high values to two consecutive points, the actual outlying point as well as the neighboring point, and therefore increases the false positive rate (because the neighboring points that are declared to be outliers actually correspond to typical points in the original data space).

Our goal was to detect suitable transformations, combinations of variables, and the algorithms for outlier score calculation for the data from two study sites. Results may depend on the characteristics of the time series (site and time dependent for example), and what is best for one site may not be the best for another site. Therefore care should be taken to select transformations most suitable for the problem at hand. According to Dang and Wilkinson (2014), any transformation used on a dataset must be evaluated

in terms of a figure of merit (i.e. a numerical quantity used to characterize the performance of a method, relative to its alternatives). For our work on detecting outliers, the figure of merit was the maximum separability of the two classes generated by outliers and typical points. However, we acknowledge that the set of transformations that we used for this work was relatively limited and influenced by the data obtained from the two study sites. Therefore, the set of transformations we considered (Table 1) should be viewed only as an illustration of our oddwater procedure for detecting outliers. We expect that the set of transformations will expand over time as the oddwater procedure is used for other data from other study sites and for applications to other fields.

For the current work we selected transformation methods that could highlight abrupt changes in the water-quality data.We hope to expand the ability of oddwater procedure so that it can detect other outlier types not previously targeted but commonly observed in water quality data (e.g.: low/high variability, drift etc. as per Leigh et al. (2019)). One possibility is to consider the residuals at each point, defined as the difference between the actual values and the fitted values (similar to Schwarz (2008)) or the difference between the actual values and the predicted values (similar to Hill and Minsker (2006)), as a transformation and apply outlier detection algorithms to the high dimensional space defined by the residuals. Here the challenge will be to identify the appropriate curve fitting and prediction models to generate the residual series. In this way, continuous subsequences of high values could correspond to other kinds of technical outliers such as high variability or drift. However, the range of applications and the space of the transformations are extremely diverse, which makes it challenging to provide a structured formal vision that covers all of the possible transformations that could be considered. The transformations we presented in this paper were mainly chosen as appropriate to the data collected from Sandy Creek and Pioneer River. We observed that different transformations can lead to entirely different data structures and that the selection of suitable transformations is directed by the data features and typical patterns imposed by a given application. Domain specific knowledge plays a vital role when selecting suitable transformations and as such defining structured guidelines for the selection of suitable transformations remains problematic.

Not surprisingly, NN-HD algorithm required the least computational time given the outlying score calculation only involves searching for the single most nearest neighbors of each test point (Wilkinson, 2018). The mean computational time of KNN-AGG was twice as high as that of KNN-SUM because the KNN-AGG algorithm has the additional requirement of calculating weights that assign nearest neighbors higher weight relative to the neighbors farther apart (Angiulli & Pizzuti, 2002). LOF and its extensions (INFLO, COF and LDOF) required the most computational time; all four algorithms involve a two step searching mechanism at each test point when calculating the corresponding outlying score. This means that at each test point each algorithm searches its k nearest neighbors as well those of the detected nearest neighbors for the outlier score calculation (Breunig et al., 2000; Tang et al., 2002; Jin et al., 2006; Zhang et al., 2009).

We hope to extend our multivariate outlier detection framework into space and time so that it can deal with the spatio-temporal correlation structure along branching river networks. Further, in the current paper we have introduced our oddwater procedure as a batch method. However, due to the unsupervised nature of our oddwater procedure it can be easily extended to a streaming data scenario with the help of a sliding window of fixed length. A streaming data scenario always demands a near-real-time support. Therefore one significant challenge is to find efficient methods that allow us to update outlier scores taking account of the newest observations and removing the oldest observations introduced by overlapping sliding windows, rather than recalculating scores corresponding to observations which are not affected by either new arrivals or the oldest observations (that are no longer covered by the latest window). Further work will be needed to

investigate the efficient computation of regenerating nearest neighbours in a data streaming context.

Acknowledgments

Funding for this project was provided by the Queensland Department of Environment and Science (DES) and the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS). The authors would like to acknowledge the Queensland Department of Environment and Science; in particular, the Great Barrier Reef Catchment Loads Monitoring Program for the data, and the staff from Water Quality and Investigations for their input. We thank Ryan S. Turner and Erin E. Peterson for several valuable discussions regarding project requirements and water quality characteristics. Further, this research was supported in part by the Monash eResearch Centre and eSolutions-Research Support Services through the use of the MonARCH (Monash Advanced Research Computing Hybrid) HPC Cluster. We would also like to thank David Hill and other anonymous reviewers for their valuable comments and suggestions. The datasets used for this article are available in the open source R package `oddwater` (Talagala & Hyndman, 2018).

References

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery* (pp. 15–27).

Archer, C., Baptista, A., & Leen, T. K. (2003). Fault detection for salinity sensors in the columbia estuary. *Water Resources Research*, *39*(3).

Ben-Gal, I. (2005). Outlier detection. In *Data mining and knowledge discovery handbook* (pp. 131–146). Springer.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: identifying density-based local outliers. In *Acm sigmod record* (Vol. 29, pp. 93–104).

Burridge, P., & Taylor, A. M. R. (2006). Additive outlier detection via extreme-value theory. *Journal of Time Series Analysis*, *27*(5), 685–701.

Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., ... Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, *30*(4), 891–927.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 15.

Dang, T. N., & Wilkinson, L. (2014). Transforming scagnostics to reveal hidden features. *IEEE transactions on visualization and computer graphics*, *20*(12), 1624–1632.

Embrechts, P., Klüppelberg, C., & Mikosch, T. (2013). *Modelling extremal events: for insurance and finance*. Springer Berlin Heidelberg. Retrieved from `https://books.google.com.au/books?id=BXOI2pICfJUC`

Gao, J., Hu, W., Zhang, Z. M., Zhang, X., & Wu, O. (2011). Rkof: robust kernel-based local outlier detection. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 270–283).

Glasgow, H. B., Burkholder, J. M., Reed, R. E., Lewitus, A. J., & Kleinman, J. E. (2004). Real-time remote monitoring of water quality: a review of current applications, and advancements in sensor, telemetry, and computing technologies. *Journal of Experimental Marine Biology and Ecology*, *300*(1-2), 409–448.

Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, *11*(4), e0152173.

Hill, D. J., & Minsker, B. S. (2006). Automated fault detection for in-situ environmental sensors. In *Proceedings of the 7th international conference on hydroinformatics.*

Hill, D. J., Minsker, B. S., & Amir, E. (2009). Real-time bayesian anomaly detection in streaming environmental data. *Water Resources Research*, *45*(4).

Hossin, M., & Sulaiman, M. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 1.

Jin, W., Tung, A. K., Han, J., & Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 577–593).

Koch, M. W., & McKenna, S. A. (2010). Distributed sensor fusion in water quality event detection. *Journal of Water Resources Planning and Management*, *137*(1), 10–19.

Kotamäki, N., Thessler, S., Koskiaho, J., Hannukkala, A. O., Huitu, H., Huttula, T., ... Järvenpää, M. (2009). Wireless in-situ sensor network for agriculture and water monitoring on a river basin scale in southern finland: Evaluation from a data users perspective. *Sensors*, *9*(4), 2862–2883.

Kriegel, H.-P., Kröger, P., & Zimek, A. (2010). Outlier detection techniques. *Tutorial at KDD*, *10*.

Leigh, C., Alsibai, O., Hyndman, R. J., Kandanaarachchi, S., King, O. C., McGree, J. M., ... others (2019). A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. *Science of The Total Environment*, *664*, 885–898.

Madsen, J. H. (2018). Ddoutlier: Distance and density-based outlier detection [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=DDoutlier` (R package version 0.1.0)

McInnes, K., Abbs, D., Bhend, J., Chiew, F., Church, J., Ekstrm, M., ... Whetton, P. (2015). *Wet tropics cluster report: Climate change in australia projections for australia's nrm regions.* CSIRO.

McKenna, S. A., Hart, D., Klise, K., Cruz, V., & Wilson, M. (2007). Event detection from water quality time series. In *World environmental and water resources congress 2007: Restoring our natural habitat* (pp. 1–12).

Mersmann, O. (2018). microbenchmark: Accurate timing functions [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=microbenchmark` (R package version 1.4-4)

Mitchell, C., Brodie, J., & White, I. (2005). Sediments, nutrients and pesticide residues in event flow conditions in streams of the mackay whitsunday region, australia. *Marine Pollution Bulletin*, *51*(1-4), 23–36.

Moatar, F., Fessant, F., & Poirel, A. (1999). ph modelling by neural networks. application of control and validation data series in the middle loire river. *Ecological Modelling*, *120*(2-3), 141–156.

Moatar, F., Miquel, J., & Poirel, A. (2001). A quality-control method for physical and chemical monitoring data. application to dissolved oxygen levels in the river loire (france). *Journal of Hydrology*, *252*(1-4), 25–36.

Panguluri, S., Meiners, G., Hall, J., & Szabo, J. (2009). Distribution system water quality monitoring: Sensor technology evaluation methodology and results. *US Environ. Protection Agency, Washington, DC, USA, Tech. Rep. EPA/600/R-09/076*, *2772*.

R Core Team. (2018). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Raciti, M., Cucurull, J., & Nadjm-Tehrani, S. (2012). Anomaly detection in water management systems. In *Critical infrastructure protection* (pp. 98–119). Springer.

Ranawana, R., & Palade, V. (2006). Optimized precision-a new measure for classifier performance evaluation. In *Evolutionary computation, 2006. cec 2006. ieee congress on* (pp. 2254–2261).

Rangeti, I., Dzwairo, B., Barratt, G. J., & Otieno, F. A. (2015). Validity and errors in water quality dataa review. In *Research and practices in water quality.* InTech.

Schwarz, K. T. (2008). *Wind dispersion of carbon dioxide leaking from underground sequestration, and outlier detection in eddy covariance data using extreme value theory.* ProQuest.

Shahid, N., Naqvi, I. H., & Qaisar, S. B. (2015). Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey. *Artificial Intelligence Review*, *43*(2), 193–228.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437.

Storey, M. V., Van der Gaag, B., & Burns, B. P. (2011). Advances in on-line drinking water quality monitoring and early warning systems. *Water research*, *45*(2), 741–747.

Talagala, P., Hyndman, R., Smith-Miles, K., Kandanaarachchi, S., & Munoz, M. (2018). *Anomaly detection in streaming nonstationary temporal data* (Tech. Rep.). Monash University, Department of Econometrics and Business Statistics.

Talagala, P., & Hyndman, R. J. (2018). oddwater: A package for outlier detection in water quality sensor data [Computer software manual]. https://github.com/pridiltal/oddwater. DOI: 10.5281/zenodo.2890469.

Tang, J., Chen, Z., Fu, A. W.-C., & Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 535–548).

Thottan, M., & Ji, C. (2003). Anomaly detection in ip networks. *IEEE Transactions on signal processing*, *51*(8), 2191–2204.

Tutmez, B., Hatipoglu, Z., & Kaymak, U. (2006). Modelling electrical conductivity of groundwater using an adaptive neuro-fuzzy inference system. *Computers & geosciences*, *32*(4), 421–433.

Weissman, I. (1978). Estimation of parameters and large quantiles based on the k largest observations. *Journal of the American Statistical Association*, *73*(364), 812–815.

Wilkinson, L. (2018). Visualizing big data outliers through distributed aggregation. *IEEE transactions on visualization and computer graphics*, *24*(1), 256–266.

Yu, J. (2012). A bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses. *Computers & Chemical Engineering*, *41*, 134–144.

Zhang, K., Hutter, M., & Jin, H. (2009). A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 813–822).

Supporting Information for "A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors"

Priyanga Dilini Talagala[1,2], Rob J. Hyndman[1,2], Catherine Leigh[1,3], Kerrie Mengersen[1,4], Kate Smith-Miles[1,5]

[1]ARC Centre of Excellence for Mathematics and Statistical Frontiers (ACEMS), Australia
[2]Department of Econometrics and Business Statistics, Monash University, Australia
[3]Institute for Future Environments, Science and Engineering Faculty, Queensland University of Technology, Australia
[4]School of Mathematical Sciences, Science and Engineering Faculty, Queensland University of Technology, Australia
[5]School of Mathematics and Statistics, University of Melbourne, Australia

Contents

Introduction

We considered the following outlier scoring techniques for the current work presented in this paper. The oddwater procedure can be easily updated with other unsupervised outlier scoring techniques.

Text S1.

NN-HD algorithm

This algorithm is inspired by the HDoutliers algorithm (Wilkinson, 2018) which is an unsupervised outlier detection algorithm that searches for outliers in high dimensional data assuming there is a large distance between outliers and the typical data. Nearest neighbor distances between points are used to detect outliers. However, variables with large variance can bring disproportional influence on Euclidean distance calculation. Therefore, the columns of the data sets are first normalized such that the data are bounded by the unit hyper-cube. The nearest neighbor distances are then calculated for each observation. In contrast to the implementation of HDoutliers algorithm available in the `HDoutliers` package (Fraley, 2018) our implementation available through the `oddwater` package now generates outlier scores instead of labels for each observation.

KNN-AGG and KNN-SUM algorithms

The NN-HD algorithm uses only nearest neighbor distances to detect outliers under the assumption that any outlying point present in the data set is isolated. For example, if there are two outlying points that are close to one another, but are far away from the rest of the valid data points, then the two outlying points become nearest neighbors to one another and give a small nearest neighbor distance for each outlying point. Because the NN-HD algorithm is dependent on the nearest neighbor distances, and the two outlying points do not show any significant deviation from other typical points with respect to nearest neighbor distance, the NN-HD algorithm now fails to detect these points as outliers.

Corresponding author: Priyanga Dilini Talagala, `dilini.talagala@monash.edu`

Following Angiulli and Pizzuti (2002), Madsen (2018) proposed two algorithms: aggregated k-nearest neighbor distance (KNN-AGG); and sum of distance of k-nearest neighbors (KNN-SUM) to overcome this limitation by incorporating k nearest neighbor distances for the outlier score calculation. The algorithms start by calculating the k nearest neighbor distances for each point. The k-dimensional tree (kd-tree) algorithm (Bentley, 1975) is used to identify the k nearest neighbors of each point in a fast and efficient manner. A weight is then calculated using the k nearest neighbor distances and the observations are ranked such that outliers are those points having the largest weights. For KNN-SUM, the weight is calculated by taking the summation of the distances to the k nearest neighbors. For KNN-AGG, the weight is calculated by taking a weighted sum of distances to k nearest neighbors, assigning nearest neighbors higher weight relative to the neighbors father apart.

LOF algorithm

The Local Outlier Factor (LOF) algorithm (Breunig et al., 2000) calculates an outlier score based on how isolated a point is with respect to its surrounding neighbors. Data points with a lower density than their surrounding points are identified as outliers. The local reachable density of a point is calculated by taking the inverse of the average readability distance based on the k (user defined) nearest neighbors. This density is then compared with the density of the corresponding nearest neighbors by taking the average of the ratio of the local reachability density of a given point and that of its nearest neighbors.

COF algorithm

One limitation of LOF is that it assumes that the outlying points are isolated and therefore fails to detect outlying clusters of points that share few outlying neighbors if k is not appropriately selected (Tang et al., 2002). This is known as a masking problem (Hadi, 1992), i.e. LOF assumes both low density and isolation to detect outliers. However, isolation can imply low density, but the reverse does not always hold. In general, low density outliers result from deviation from a high density region and an isolated outlier results from deviation from a connected dense pattern. Tang et al. (2002) addressed this problem by introducing a Connectivity-based Outlier Factor (COF) that compares the average chaining distances between points subject to outlier scoring and the average of that of its neighboring to their own k-distance neighbors.

INFLO algorithm

Detection of outliers is challenging when data sets contain adjacent multiple clusters with different density distributions (Jin et al., 2006). For example, if a point from a sparse cluster is close to a dense cluster, this could be misclassified as an outlier with respect to the local neighborhood as the density of the point could be derived from the dense cluster instead of the sparse cluster itself. This is another limitation of LOF (Breunig et al., 2000). The Influenced Outlierness (INFLO) algorithm (Jin et al., 2006) overcomes this problem by considering both the k nearest neighbors (KNNs) and reverse nearest neighbors (RNNs), which allows it to obtain a better estimation of the neighborhood's density distribution. The RNNs of an object, p for example, are essentially the objects that have p as one of their k nearest neighbors. Distinguish typical points from outlying points is helpful because they have no RNNs. To reduce the expensive cost incurred by searching a large number of KNNs and RNNs, the kd-tree algorithm was used during the search process.

LDOF algorithm

The Local Distance-based Outlier Factor (LDOF) algorithm (Zhang et al., 2009) also uses the relative location of a point to its nearest neighbors to determine the degree to which the point deviates from its neighborhood. LDOF computes the distance for an observation to its k-nearest neighbors and compares the distance with the average distances of the point's nearest neighbors. In contrast to LOF (Breunig et al., 2000), which uses local density, LDOF now uses relative distances to quantify the deviation of a point from its neighborhood system. One of the main differences between the two approaches (LDOF and LOF) is that LDOF represents the typical pattern of the data set by scattered points rather than crowded main clusters as in LOF (Zhang et al., 2009).

RKOF algorithm with Gaussian kernel

According to Gao, Hu, Zhang, Zhang, and Wu (2011), LOF is not accurate enough to detect outliers in complex and large data sets. Furthermore, the performance of LOF depends on the parameter k that determines the scale of the local neighborhood. The Robust Kernel-based Outlier Factor (RKOF) algorithm (Gao et al., 2011) tries to overcome these problems by incorporating variable kernel density estimates to address the first problem and weighted neighborhood density estimates to address the second problem. A Gaussian kernel with a bandwidth of $k-distance$ was used for density estimation. The two parameters: multiplication parameter for $k-distance$ of neighboring observations and sensivity parameter for $k-distance$ were set to 1 (default value given in Gao et al. (2011)).

References

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery* (pp. 15–27).

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, *18*(9), 509–517.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: identifying density-based local outliers. In *Acm sigmod record* (Vol. 29, pp. 93–104).

Fraley, C. (2018). Hdoutliers: Leland wilkinson's algorithm for detecting multidimensional outliers [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=HDoutliers` (R package version 1.0)

Gao, J., Hu, W., Zhang, Z. M., Zhang, X., & Wu, O. (2011). Rkof: robust kernel-based local outlier detection. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 270–283).

Hadi, A. S. (1992). Identifying multiple outliers in multivariate data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 761–771.

Jin, W., Tung, A. K., Han, J., & Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 577–593).

Madsen, J. H. (2018). Ddoutlier: Distance and density-based outlier detection [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=DDoutlier` (R package version 0.1.0)

Tang, J., Chen, Z., Fu, A. W.-C., & Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 535–548).

Wilkinson, L. (2018). Visualizing big data outliers through distributed aggregation.

IEEE transactions on visualization and computer graphics, *24*(1), 256–266.

Zhang, K., Hutter, M., & Jin, H. (2009). A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 813–822).